# Assignment 9: Applications of Search

### Overview

The purpose of this activity is to apply what you've learned about Problem Solving via Search to a real-world problem. You'll start with a problem accompanied by some data. Your job is to outline how to construct a solution to this problem using the techniques discussed in class (and applied in the sudoku assignment).

### Problem

In cohorted classes, students want to work with as many different partners as possible, while balancing skill levels in the relevant domain. Students would like a "random" new partner for each assignment, while also maintaining fair teams where experts are paired with newbies.  Unfortunately, doing this task by hand is very tedious. As we get further into the quarter, it becomes more difficult to make a unique set of partnerships.

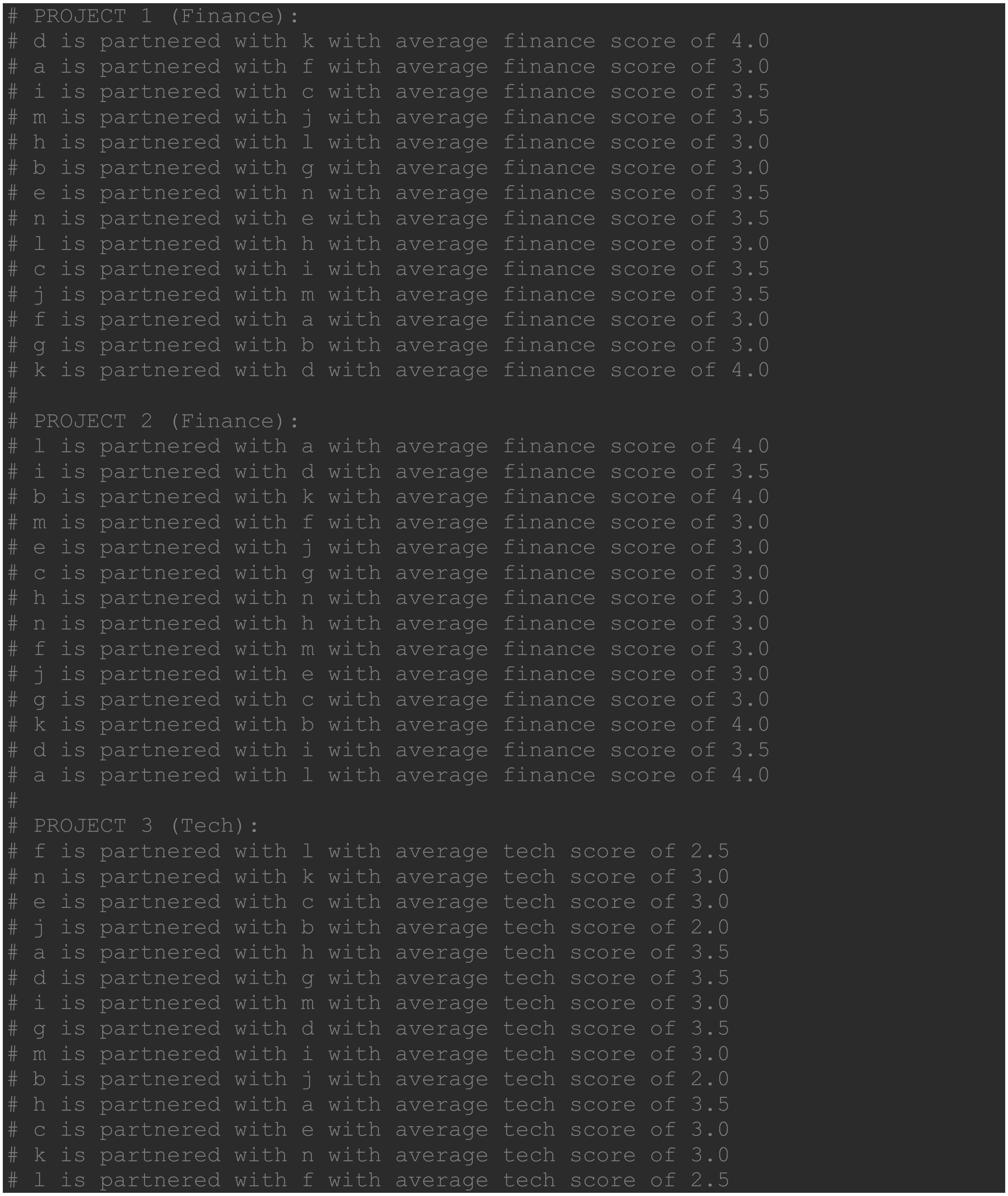
### Your job

Imagine that you have a text file, called cohort.txt, that contains information about students in the cohort. Each line of the file contains (self-reported) data about one person in the following format:  name, finance score, tech score
For example,

```
name, finance_score, tech_score
a, 4, 2
b, 5, 1
c, 5, 3
d, 5, 2
e, 3, 3
f, 2, 4
g, 1, 5
h, 2, 5
i, 2, 3
j, 3, 3
k, 3, 2
l, 4, 1
m, 4, 3
n, 4, 4
```

We want to construct a system that will take this input file, containing information about the cohort, and generate *n* unique pairings – *f* of which are varied in finance skills and *t* of which are varied in technical skills. For example, generate_pairings("cohort.txt", 3, 2, 1) would generate pairings of students for 3 unique projects – 2 for finance projects and 1 for a tech project, using cohort.txt as a source of student data.

Here is a valid sample output given the above call…



```
# PROJECT 1 (Finance):
# d is partnered with k with average finance score of 4.0
# a is partnered with f with average finance score of 3.0
# i is partnered with c with average finance score of 3.5
# m is partnered with j with average finance score of 3.5
# h is partnered with l with average finance score of 3.0
# b is partnered with g with average finance score of 3.0
# e is partnered with n with average finance score of 3.5
# n is partnered with e with average finance score of 3.5
# l is partnered with h with average finance score of 3.0
# c is partnered with i with average finance score of 3.5
# j is partnered with m with average finance score of 3.5
# f is partnered with a with average finance score of 3.0
# g is partnered with b with average finance score of 3.0
# k is partnered with d with average finance score of 4.0
#
# PROJECT 2 (Finance):
# l is partnered with a with average finance score of 4.0
# i is partnered with d with average finance score of 3.5
# b is partnered with k with average finance score of 4.0
# m is partnered with f with average finance score of 3.0
# e is partnered with j with average finance score of 3.0
# c is partnered with g with average finance score of 3.0
# h is partnered with n with average finance score of 3.0
# n is partnered with h with average finance score of 3.0
# f is partnered with m with average finance score of 3.0
# j is partnered with e with average finance score of 3.0
# g is partnered with c with average finance score of 3.0
# k is partnered with b with average finance score of 4.0
# d is partnered with i with average finance score of 3.5
# a is partnered with l with average finance score of 4.0
#
# PROJECT 3 (Tech):
# f is partnered with l with average tech score of 2.5
# n is partnered with k with average tech score of 3.0
# e is partnered with c with average tech score of 3.0
# j is partnered with b with average tech score of 2.0
# a is partnered with h with average tech score of 3.5
# d is partnered with g with average tech score of 3.5
# i is partnered with m with average tech score of 3.0
# g is partnered with d with average tech score of 3.5
# m is partnered with i with average tech score of 3.0
# b is partnered with j with average tech score of 2.0
# h is partnered with a with average tech score of 3.5
# c is partnered with e with average tech score of 3.0
# k is partnered with n with average tech score of 3.0
# l is partnered with f with average tech score of 2.5
```

Your job is to formulate this as a search problem. That is, how will you express the following….

- initial state,
- goal test,
- failure test (optionally) and

- actions/operations?

Which search strategy will you use and why?

Feel free to freely outline (pencil/paper or whiteboard or code) solutions to this problem. If you want some more guidance, here are some things to consider discussing.

1) How will you represent each student? What data do you need to store about each student? Design a *student* class. Outline the data members and member functions.
2) How will you represent the pairings for a single project? Design a *pairings* class. What are the data members and member functions?
3) Assuming that pairings is intended as a "state" in our problem-solving system, what does the *initial state* look like? Is it a *complete* or *partial information* state?
4) What is an *action/operation* on a state?
5) What does a *goal test* look like?
6) What does a *failure test* look like?
7) Which search strategy will you use and why? Do you anticipate any challenges with regards to time, space, completeness or optimality?

What to turn in? Anything you produce in the above, this could be a code file or a simple word document formulating the problem. Please note, to get full credit for this assignment, you only need to sketch out a solution. Implementation is completely optional.